Listing ofClaims

1. (Currently Amended) A PLL/DLL dual loop data serializer comprising:

a phase lock loop (PLL) including,

a phase frequency detector (PFD) receiving a local clock,

a voltage controlled oscillator (VCO),

a loop filter coupled to said PFD and to said VCO, said loop filter configured to suppress VCO phase noise, and

a phase shifter coupled to said VCO and configured in a feedback loop with said PFD;

a delayed lock loop (DLL) having a digital loop filter coupled to a phase detector and to said phase shifter of said PLL;

a first-in first-out (FIFO) register receiving a parallel data input and outputting a signal to said phase detector; and

a parallel-in serial- out (PISO) serializer receiving an input from said FIFO and outputting serialized data.

2. (Original) The data serializer of claim 1, wherein said DLL is embedded in said PLL.

3. (Original) The data serializer of claim 1, wherein said PLL locks to said signal from said FIFO to said phase detector.

4. (Original) The data serializer of claim 1, wherein said loop filter of said PLL comprises a wideband filter.

5. (Original) The data serializer of claim 1, wherein said loop filter of said DLL comprises a narrowband filter.

6. (Original) The data serializer of claim 1, wherein said signal to said phase detector comprises a FIFO fill level indicator.

7. (Original) The data serializer of claim 6, wherein said phase detector is configured to translate said FIFO fill level into a digital value.

8. (Original) The data serializer of claim 1 for use in a plesiochronous system.

9. (Original) A dual loop retimer comprising the data serializer of claim 1.

10. (Currently Amended) A method for PLL/DLL data serialization comprising:

detecting a local reference at a phase/frequency detector (PFD) of a phase lock loop (PLL);

phase locking a voltage controlled oscillator (VCO) of said PLL to a local reference to suppress a phase noise of said VCO;

receiving a parallel data input and a data clock at a FIFO register;

filtering, at a delayed lock loop (DLL), a signal representative of ~~said~~ a fill level of said FIFO;

phase shifting an output of said VCO of said PLL in response to said filtering step;

locking said PLL to a frequency corresponding to ~~said~~ a pre-filtered signal input to said DLL;

receiving, at a parallel-in serial-out (PISO) serializer, said parallel data and said VCO output; and

outputting a serialized data from said PISO serializer with said VCO output a transmit clock.

11. (Original) The method of claim 10, further comprising the step of outputting a synthesized clock.

12. (Original) The method of claim 10, wherein said PLL filtering step comprises wide bandwidth filtering.

13. (Original) The method of claim 10, wherein said DLL filtering step comprises narrow bandwidth filtering.

14. (Original) The method of claim 10, further comprising the step of translating said signal in said DLL to a digital value.

15. (Original) The method of claim 14, wherein said translating step comprises a phase detector in said DLL.

16. (Original) The method of claim 10 comprising a plesiochronous system.

17. (Original) A plesiochronous data retimer comprising:

a digital delay lock loop (DDLL) receiving an input data to be retimed and configured to recover a clock of said input data;

a dual loop serializer having a phase lock loop (PLL) and a delay lock loop (DLL), said serializer comprising;

a phase/frequency detector (PFD) receiving a local reference at said PLL,

a phase shifter configured in a feedback loop with said PFD within said PLL;

a loop filter within said DLL and coupled to said phase shifter;

a SIPO (serial-in and parallel-out) deserializer coupled to said input data;

a FIFO register coupled to said deserializer and said serializer DLL; and

a PISO (parallel-in and serial-out) serializer receiving said deserialized input data and transmitting a serialized data.

18. (Original) The retimer of claim 17, wherein said DDLL comprises a phase detector and a digital loop filter.

19. (Original) The retimer of claim 18, wherein said DDLL comprises a wide bandwidth.

20. (Original) The retimer of claim 18, wherein said serializer further comprises a loop filter within said PLL.

21. (Original) The retimer of claim 20, wherein said serializer comprises a dual bandwidth.

22. (Original) The retimer of claim 20, wherein said DLL loop filter comprises a narrow bandwidth and said PLL loop filter comprises a wide bandwidth.

23. (Previously presented) A plesiochronous data retiming method comprising:

recovering a clock from a received serial input data at a digital delay locked loop (DDLL);

deserializing said serial data to a parallel data using said recovered clock;

writing said parallel data to a FIFO (first-in first-out);

synthesizing a transmit clock;

reading said parallel data from said FIFO;

serializing said parallel data using said synthesized transmit clock;

detecting a FIFO fill level at a delay locked loop (DLL); and

phase shifting, in a phase lock loop (PLL), an output of a VCO, wherein said phase shifting is in response to said detecting step.

24. (Original) The retiming method of claim 23, wherein said synthesizing step comprises phase locking said VCO to a local reference.

25. (Original) The retiming method of claim 23, further comprising translating said FIFO fill level to an integrating value.

26. (Original) The retiming method of claim 23, wherein said writing step and said reading step comprise a write clock of said FIFO and a read clock of said FIFO, respectively.

27. (Original) The retiming method of claim 26, further comprising phase locking said write and read clocks of said FIFO in said DLL.

28. (Original) The retiming method of claim 26, further comprising locking said VCO output to said FIFO write clock.

29. (Original) A method for PLL/DLL data retiming comprising:

recovering a clock from a received serial input data at a digital delay locked loop (DDLL);

writing said serial data to a FIFO (first-in first-out);

synthesizing a transmit clock;

reading a retimed data from said FIFO;

detecting a FIFO fill level at a delay locked loop (DLL); and

phase shifting, in a phase lock loop (PLL), an output of a VCO, wherein said phase shifting is in response to said detecting step.

30. (Original) The method of claim 29, further comprising the step of outputting a synthesized clock.

31. (Original) The method of claim 29, further comprising the step of phase locking said VCO of said PLL to a local reference to suppress a phase noise of said VCO.

32. (Original) The method of claim 31, wherein said PLL filtering step comprises wide bandwidth filtering.

33. (Original) The method of claim 29, further comprising the step of translating said FIFO fill level to an integrating value.

34. (Original) The method of claim 33, wherein said translating step comprises a phase detector in said DLL.

35. (Original) The method of claim 29 comprising a plesiochronous system.